

Agenda StUF Expertgroep

Datum	Woensdag 21 januari 2009
Locatie	Vergadercentrum Hoog Brabant, Utrecht
Tijd:	9.30 uur – 12.30 uur

1. Opening en mededelingen
2. Notulen vorige vergadering en actielijst
3. Bespreken feedback Patch StUF-BG 02.04
4. Hoe om te gaan met restriction en namespaces (4 opties van Maarten, zie onderstaande tekst)
5. Bespreken nieuwe versie 'Ontwerpkeuzen bij het verStUFfen van het RGSB'
6. Bespreken nieuwe XML schema's voor sectormodel StUF-BG 3.10
7. Rondvraag en sluiting

Ad 4)

In het StUF-raamwerk correspondeert een sectormodel met een namespace (die per versie van het sectormodel kan verschillen). Het sectormodel BG (basisgegevens) bevat objecttypen die ook binnen andere sectormodellen gebruikt worden en vaak zelfs met meer of minder attributen en relaties. Denk bijvoorbeeld aan het WOZ-object en het WOZ-deelobject die in het WOZ-domein veel meer attributen hebben dan in het RGSB of aan Natuurlijk Persoon en Kadastrale Onroerende Zaak die in het WOZ-domein minder attributen hebben.

In het denken over een en ander speelt op de achtergrond de 'oude' eis, dat een StUF-verwerker elementen die het niet kent in een bericht dient te negeren, zelfs als het bericht daardoor niet valideert. Dit was een zinnige eis toen er nog geen schema's waren. In XML-technologie is deze eis echter achterhaald: een bericht dat niet valideert is ongeldig.

Je kunt met deze problematiek op de volgende manieren omgaan:

1. Restrict/extend het complexType in de oorspronkelijke namespace. Aan de oorspronkelijke namespace worden in geval van extension hierdoor buiten de beheerder van deze namespace om nieuwe complexTypes toegevoegd. Met deze werkwijze zitten alle elementen van het uitgebreidere complexType in de oorspronkelijke namespace. Dit is de werkwijze die ik tot op heden heb gevolgd. Gegeven de 'oude' eis zou een WOZ-object aangemaakt in het WOZ-domein foutloos verwerkt moeten kunnen worden door StUF-verwerker die werkt conform het sectormodel BG.
2. Kopieer wat je nodig hebt naar de nieuwe namespace en voeg vervolgens toe wat je nodig hebt. Ook in dit geval zitten alle elementen in dezelfde namespace, maar zijn de dubbel voorkomende elementen twee maal gedefinieerd. De vraag is of het in dit geval wenselijk is om exact dezelfde volgorde van elementen etc. te handhaven.
3. Als 2, maar gebruik het substitutionGroup mechanisme om aan te geven dat elementen in de nieuwe namespace een substitution zijn voor elementen in de oorspronkelijke namespace. Deze variant is primair cosmetisch en heeft als nadeel dat in BG alle elementen binnen de complexTypes voor entiteitstypen als global element gedefinieerd moeten worden.

4. Restrict in de oorspronkelijke namespace een complexType tot wat je ervan wilt gebruiken in de nieuwe namespace, maar doe uitbreidingen met nieuwe elementen in de nieuwe namespace. Deze variant is conceptueel het zuiverste, omdat het nieuwe sectormodel geen uitbreidingen doet op een al bestaande namespace. Het leidt er wel toe dat binnen één complexType voor een entiteitstype elementen uit verschillende namespaces voorkomen.

Het lijkt me goed om deze discussie op korte termijn te voeren in de expertgroep. Met de kennis die ik nu heb, gaat mijn voorkeur uit naar de laatste variant. In de laatste variant is het eenvoudig om met xslt een element voor een WOZ-object in het sectormodel WOZ te transformeren naar het corresponderende element in het sectormodel BG. Een mooie taak voor een servicebus.

Voor het maken van het sectormodel WOZ dienen we een keuze gemaakt te hebben.